Eur. Phys. J. C (2021) 81:599
https://doi.org/10.1140/epjc/s10052-021-09366-4

THE EUROPEAN PHYSICAL JOURNAL C

Check for updates

Regular Article - Experimental Physics

Simulating the time projection chamber responses at the MPD detector using generative adversarial networks

A. Maevskiy[1,a], **F. Ratnikov**[1,2,b], **A. Zinchenko**[3,c], **V. Riabov**[4,d]

[1] HSE University, 20 Myasnitskaya Ulitsa, Moscow, Russia
[2] Yandex School of Data Analysis, 11-2 Timura Frunze Street, Moscow, Russia
[3] Joint Institute for Nuclear Research, 6 Joliot-Curie St, Dubna, Moscow Oblast, Russia
[4] Petersburg Nuclear Physics Institute, 1, mkr. Orlova roshcha, Gatchina, Leningradskaya Oblast, Russia

Received: 13 December 2020 / Accepted: 23 June 2021 / Published online: 10 July 2021
© The Author(s) 2021

Abstract High energy physics experiments rely heavily on the detailed detector simulation models in many tasks. Running these detailed models typically requires a notable amount of the computing time available to the experiments. In this work, we demonstrate a new approach to speed up the simulation of the Time Projection Chamber tracker of the MPD experiment at the NICA accelerator complex. Our method is based on a Generative Adversarial Network – a deep learning technique allowing for implicit estimation of the population distribution for a given set of objects. This approach lets us learn and then sample from the distribution of raw detector responses, conditioned on the parameters of the charged particle tracks. To evaluate the quality of the proposed model, we integrate a prototype into the MPD software stack and demonstrate that it produces high-quality events similar to the detailed simulator, with a speed-up of at least an order of magnitude. The prototype is trained on the responses from the inner part of the detector and, once expanded to the full detector, should be ready for use in physics tasks.

1 Introduction

Computer simulations of high-energy physics experiments play a crucial role in a variety of relevant tasks, including detector geometry optimization [1,2], selecting best analysis strategies [3,4], and testing the Standard Model (SM) predictions and searching for new phenomena beyond the SM [5,6]. For a typical experimental data analysis, the number of simulated events usually translates directly to the uncertainty of the final physics result. The amount of computational resources spent on the simulations usually takes a notable fraction of the total computing capabilities of an experiment and is comparable with that spent on the real data processing [7,8]. Therefore, faster approaches to event generation and simulation are in great demand for the existing and future high energy physics experiments.

The MPD detector is one of the two experiments at the NICA accelerator complex – a new heavy ion accelerator facility being constructed at the Joint Institute for Nuclear Research and located in Dubna, Russia [9,10]. The complex is designed to study the properties of dense baryonic matter. For the tracking, MPD utilizes a time projection chamber (TPC) in the central barrel [11]. TPC simulation is very CPU-intensive [12], and hence a fast simulation approach for TPC is highly desirable.

A typical approach to constructing models for fast simulation of particle physics detectors is to use a simplified detector geometry and a simplified model of the interaction of particles with matter [13]. This approach is justified for subsystems with a flat sensitive volume, such as silicon trackers, that measure the two-dimensional coordinate of a passing particle. For systems with a large volume, such as calorimeters or TPC-based trackers, this approach makes it difficult to achieve a reasonable compromise between accuracy and simulation speed.

Another fast simulation approach is an analytical parameterization of the detector responses, as can be seen in shower shape parameterizations for calorimeters [14]. This approach can significantly speed up the calorimeter simulation, but it makes it difficult to achieve high quality simulated data. A common solution for calorimeters is also to use the so-called "frozen showers" [13] when detailed simulated system responses are stored as a response library for subsequent reuse.

[a] e-mail: artem.maevskiy@cern.ch (corresponding author)
[b] e-mail: fedor.ratnikov@cern.ch
[c] e-mail: alexander.zinchenko@jinr.ru
[d] e-mail: riabovvg@gmail.com

With recent developments and particular success of deep learning in high energy physics [15–19], fast simulation models based on a Generative Adversarial Network (GAN) [20] and Variational Auto-Encoders (VAE) [21,22] have emerged (see e.g. [8,23–35]). These techniques allow to learn the data distribution, with sampling being as fast as a single forward pass through the neural network. In this work, we focus on using GANs for fast simulation. GAN training is based on a competition between two independent neural networks, the generator and discriminator. The generator aims to convert samples from a fixed known distribution into the objects from the target distribution, – the one that the training data follows. The discriminator network takes the examples from the target and generator output distributions and predicts the probability for each of these examples to belong to the target distribution. The objective of the discriminator is to learn a metric that maximizes the separation between the training data and the generated objects, while that of the generator is to minimize this separation. In [20] it is shown that the equilibrium state of such a system is a situation where the objects generated by the generator are indistinguishable from those of the training sample, and hence the generator has learned the training data distribution.

In this work, we propose a new method for fast simulation of TPC trackers applied to the tracker of the MPD detector at the NICA accelerator complex, using a GAN.

The structure of the paper is the following. In Sect. 2, previous research in applying deep generative models to the fast simulation of particle physics detectors is discussed. In Sect. 3, we describe the TPC detector at the MPD experiment. Section 4 explains our approach to simulation of the TPC with GANs. In Sect. 5, we demonstrate our results and evaluate their quality. Section 6 is dedicated to the discussion of the perspectives and limitations of our approach. Finally, Sect. 7 contains a short summary of the work.

2 Related work

As was discussed in Sect. 1, GANs have previously been applied as a tool for fast simulation of high energy physics experiments [8,23–35]. This idea is first proposed in [23], and then further developed in [24], where it is shown that GANs can significantly speed up electromagnetic calorimeter simulation by generating raw calorimeter readouts. In [27], this approach is further developed for the case of the LHCb calorimeter, and in [28] – for the ATLAS calorimeter. Calorimeters from high energy physics experiments are particularly appealing to simulate using GANs. Their responses form 2- or 3-dimensional structures of fixed size, which are similar to regular images, where GANs have demonstrated outstanding performance over the past years [36]. Use of GANs for simulation of the reconstructed characteristics is proposed for the case of Cherenkov detectors in [30], and further developed for LHCb RICH detectors in [31]. Simulation of reconstructed objects has also been proposed in [37]. In [29], GANs are used for simulating hadronic jets, as simulated and reconstructed in the CMS detector. This research is not limited to the collider experiments alone, however. E.g. in [25] GANs are used for simulating a ground-based array of particle detectors for cosmic rays.

Using GANs for the fast simulation of the TPC type detector has been studied for the ALICE experiment [32]. The main idea in that work is to use a GAN to model the measured cluster coordinates, conditioned on the charged particle track parameters. With such an approach, the size of the target representation, i.e. the number of measured track interaction points, varies from track to track. This is problematic to model with simple feed-forward or convolutional neural network architectures and typically requires more computationally expensive and harder to train techniques. The authors of [32] overcome this problem by fixing the size of the target representation to the maximum possible number of points per track and zero-padding the trajectories that have fewer points. The quality of the generated data is measured by the mean squared distance between the generated points and the true track helix. It is shown, that the proposed model cannot yet match the quality of the training data, although it accelerates the detailed simulation by a factor of 25 [32].

The idea of utilizing GANs for simulating TPC response at the MPD detector has previously been proposed in [12]. In our work, we develop upon that concept to model the raw TPC readout electronics response, rather than the measured track coordinates as done in [32]. This allows us to utilize the translational symmetries of the detector to reduce the dimensionality of the learned representation, and also to avoid the problem of the variable number of points per track. Another difference from [32] is that we condition the generation process on the parameters of the small track segments that contribute to the simulated response, rather than the parameters of the track at the production point. This means that our method can be used for events with any topology, including e.g. the effects of decay-in-flight, when a particle decays within the detector volume. This is opposed to the approach from [32] where only long enough tracks are considered and taking such effects into account would require training the model specifically for such events.

We would like to finalise this section with a short note on the applicability of GAN usage for fast detector simulation. As it is raised in [38], for a physics analysis, the systematic uncertainties associated with a GAN-generated sample cannot be smaller compared to those from the sample that GAN is trained on. At the same time, applying GANs also means making use of the prior knowledge about the structure of the data [39]. In other words, the generated data does not contain more information about the true distribution than there

is in the training sample combined with the prior knowledge introduced by the network architecture. One should note, that the same argument applies to other fast simulation techniques like analytical parameterizations or memorized event libraries. Our position is that deep generative modeling should rather be considered as a method for memorization and interpolation of the training data, and it should be applicable in those cases where analytical parameterizations and memorized event libraries are.

3 TPC tracker of the MPD experiment

TPC is the main tracking detector of the central barrel of the MPD experiment [40], covering the pseudorapidity region of $|\eta| < 1.2$ and 2π in azimuthal angle. It is designed to provide the high efficiency of the charged particle track reconstruction with a momentum resolution of better than 3% in the transverse momentum range $0.1 < p_T < 1\,\text{GeV/c}$. The double-track resolution should not exceed 1 cm for operation in a high-multiplicity environment realized in central collisions of heavy ions. Moreover, the TPC is used as one of the primary particle identification detectors. The particle identification is based on the measurement of the ionization losses with a resolution better than 8%.

The gas volume of the detector occupies the central barrel radial region between 34 and 134 cm and has a length of 3.4 m in the beam direction (z-axis). The detector is aligned along the beam axis and is centered with respect to the nominal collision point. The gas volume of the detector is enclosed in an electric field cage, which together with the central membrane effectively divides it into two identical halves with the electric field lines parallel to the beam axis. The primary ionization electrons drift towards the edges, and the signals are amplified and registered with the proportional chambers with the cathode readout. The chambers are arranged in 12 sectors on each side of the detector with each sector covering 30° in azimuth. The cathode planes of the chambers are pad-segmented with 53 pad rows perpendicular to the radial direction and the number of pads in each row increasing with the radius. The pads are 5 mm wide, while their height is 12 mm (short pads) and 18 mm (long pads) for the 27 inner and 26 outer rows, respectively. Overall, the detector has 95,232 sensitive pads, responses from which are collected in 310 time buckets per bunch crossing. The XY-coordinates of the track segments are obtained from the location of the fired pads. The z-coordinate is calculated based on the measured drift time and the known drift velocity of electrons in the gas.

4 Model description

As was mentioned in Sect. 2, we build our model to generate raw TPC responses, i.e. the responses from the sensitive pads. Given the total number of pads in TPC and the number of time buckets per event their response is recorded in, this means that we need to generate almost 30 million numbers per bunch crossing. To reduce the number of dimensions of the target space, we split the pads into smaller conditionally independent subsets, with conditions imposed by the track parameters, as described below.

In order to apply this reduction, the major assumption we make is that the response at a particular pad row depends only on small segments of the particle trajectories, formed by tracks crossing the corresponding *pad plane*. The *pad plane* is the volume swept by the electric field lines directed at the given pad row. With this assumption, we ignore the effects of electron drift diffusion in the direction orthogonal to the pad plane or the spread of the induced signal over several pad rows. To fix the dimensionality of the input space, we model the contributions from different track segments separately, combining them in an additive manner and ignoring any nonlinear effects. Later we show that the assumptions described here do not significantly affect the quality of the generated events.

Finally, we utilize the fact that responses from a particular track segment are localized in space and time, and therefore we only model a small number of pads and time buckets for a given track segment. Since the pads are identical, we only train our model on the responses from a small subset of pads in a single pad row and translate the predictions onto all other pads. In fact, there are two shapes of pads (short and long), so a more precise simulation of their responses requires either two separate models or an input condition variable specifying the type of the pad to model the response for. In this work, we train our model only on the responses from the short pads.

4.1 Data

The training dataset is obtained by running the detailed MPD simulator [41], which is based on Geant3 transport of particles through the detector materials and realistic simulation of the detector responses based on the first principles. Each of the simulated events has only one positively charged pion with the transverse momentum of $p_T = 478.3\,\text{MeV/c}$. The pions are generated uniformly along the drift path with the azimuthal angle φ in a range $[-20; 20]°$ and the polar angle θ in a range $[30, 150]°$. The generated responses are picked up from the 20th pad row for further analysis.

In the coordinate system local to a given sector of the detector, the track segment crossing a particular pad plane is defined by four parameters:

– *crossing angle* the angle between the transverse projection of the particle momentum and the normal to the pad plane;

– *dip angle* the angle between the full momentum and its transverse projection;
– *drift length* distance from the center of the segment to the triggered pad row, measured in the number of time buckets from the bunch crossing to the pad response generation;
– *pad coordinate* coordinate along the pad row direction of the projection of the track segment center onto the triggered pad row, measured in pad widths.

For the sector selected for training, the crossing angle equals the azimuthal angle φ, and the dip angle equals to $(\theta - 90)^\circ$. Overall, 20,000 segment responses were generated, which were split into training and validation subsets as 75:25, respectively.

4.2 Data preprocessing

As was mentioned earlier, pad response contributions from a single track segment affect only a few pads and time buckets. For each segment, in order to make use of this response localization, we shift the responses by the integer parts of the drift length and pad coordinate along the time and pad row directions, respectively. After having done this, the responses in the whole training set fit onto a matrix of 8 pads by 16 time buckets, which constitutes our target space. The responses span over several orders of magnitude, so we scale them with $\log_{10}(x + 1)$ for smoother learning.

Since we utilize the invariance of the responses under translations along the transverse plane (and along the pad row direction in particular), it is sufficient to only feed the fractional part of the pad coordinate into our model, rather than the full pad coordinate. As for the drift length, the response characteristics do depend on its absolute value (e.g. the spread of the response in time buckets increases with the drift length due to diffusion effects), so both the fractional part and the full number are fed into the model as two separate features. Providing the fractional part as a separate input is motivated by the fact that the invariance under longitudinal translations does hold to some degree, and, therefore, this additional feature contains meaningful information. Before being fed into the model, the angles and the drift length are linearly scaled down to a $[-1, 1]$ region.

4.3 Network architecture and the objective function

Since our target space has image-like structure, as shown in Fig. 1, it is natural to apply convolutional neural network architectures. However, we obtain that, for our data, the same generated data quality can be achieved with a fully-connected generator architecture that runs much faster compared to a convolutional network (on a single CPU). Therefore, we decide to only use convolutions in the discriminator network, where high performance is not crucial, while switch to a fully-connected network for the generator. Overall, the structures of both networks are given in Appendix A.

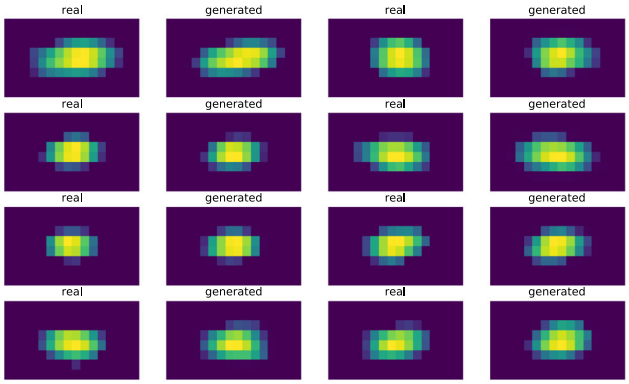


Fig. 1 Examples of the generated pad responses. Vertical and horizontal axes correspond to the pad and time bins, respectively. Each image from the validation dataset (1st and 3rd columns) is paired up with a generated image (2nd and 4th columns) obtained for the same values of the conditional variables

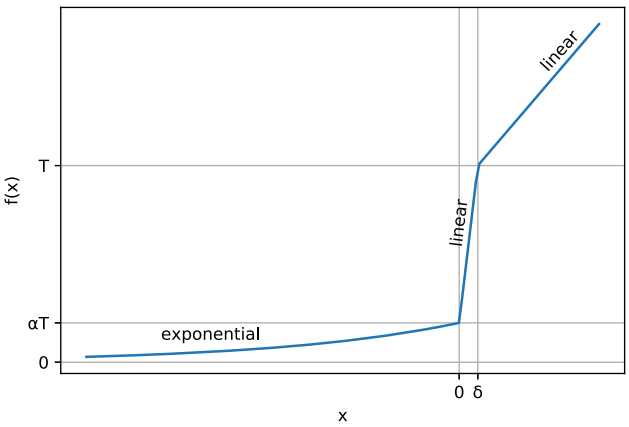


Fig. 2 Activation function $f(x)$ as defined in Eq. (1)

In order to provide the features to the discriminator, we tile each of them onto 8×16 matrices to concatenate with the main pad response image along the channels axis. Additionally, we concatenate the features vector to the dense representation obtained at the end of the convolutional part of the network. Alternative architectures, where only one of these two feature paths is kept, demonstrate inferior quality of the generated samples.

In the detailed simulator, the spectrum of the individual pad responses is continuous only down to a certain noise threshold level, below which everything is set to be exactly 0. This means that the target space we are simulating with a GAN contains a discrete mode corresponding to these zero values, which is problematic to learn. In fact, we observe that even the quality of the samples generated above the threshold

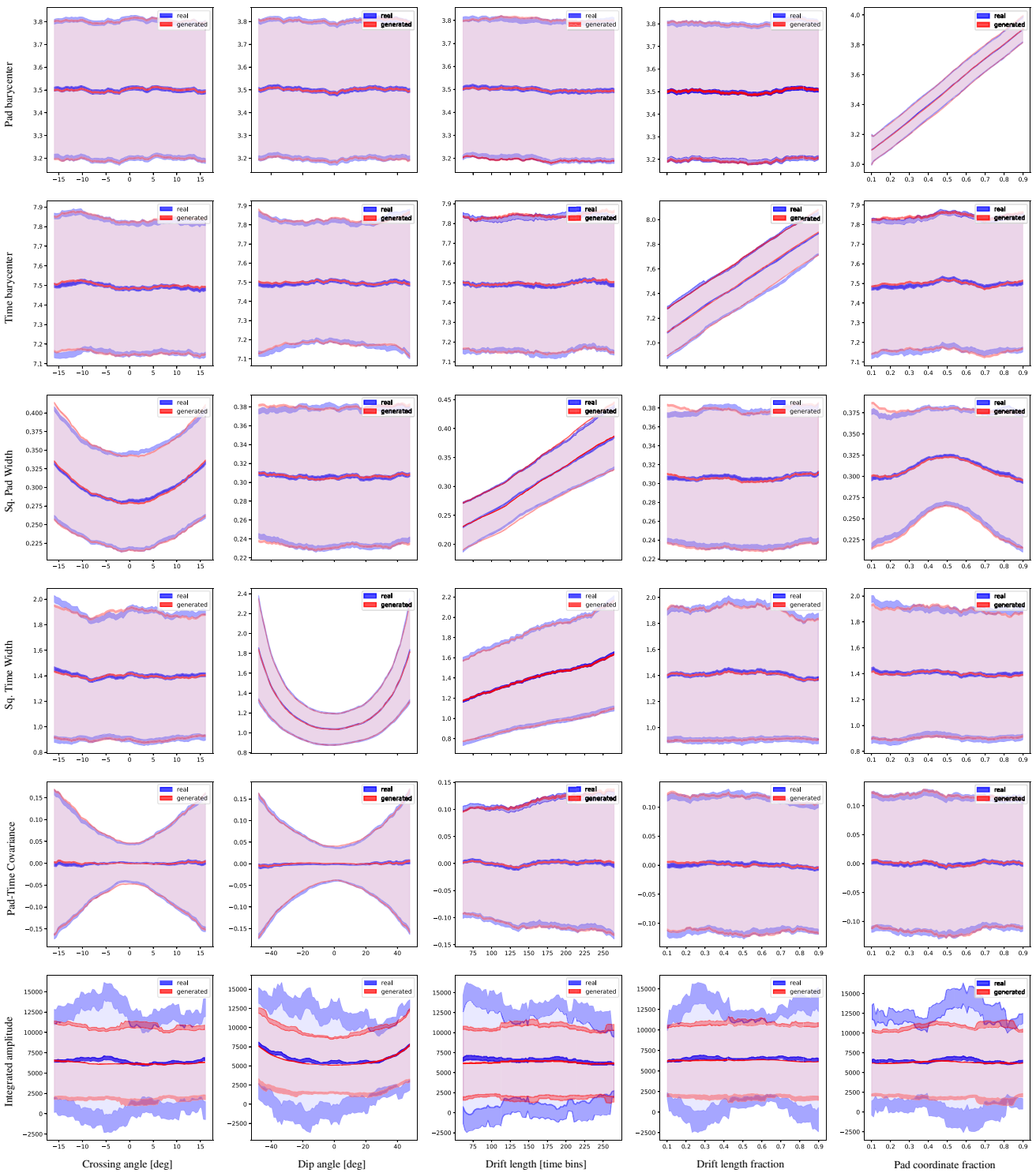


Fig. 3 Profiles of the validation metric distributions as the function of the input variables. For each metric ξ, we show its average value μ_ξ (middle thick line), as well as the average shifted up and down by a standard deviation of the metric distribution $\mu_\xi \pm \sigma_\xi$ (top and bottom thick lines). Line thickness denotes the statistical uncertainty of the corresponding value. For a smoother representation, the values were calculated in overlapping running windows over the input variables (100 bins total, 20-bin window size)

deteriorates. We attribute this effect to the GAN attempting and failing to describe the steep cutoff with its continuous output. To mitigate this problem, we use a custom activation function at the generator output layer. The function has a very steep slope for the values mapped to the interval between 0 and the threshold, by which we effectively reduce the probability of outputs to end up in that region. The particular form of the function is given by the formula (see Fig. 2):

$$f(x) = \begin{cases} \alpha T e^{x} & x \leq 0 \\ T\left(\alpha + (1-\alpha)\frac{x}{\delta}\right) & 0 \leq x \leq \delta \\ T - \delta + x & \delta \leq x, \end{cases} \quad (1)$$

where $T = \log_{10} 2$ is the threshold, $\alpha = 0.1$ and $\delta = 0.01$. We observe that using this activation function at the output layer of the generator improves the overall quality of the generated samples.

For the GAN objective, we use the Wasserstein distance [42] with the gradient penalty term from [43], as we find it resulting in the best generated data quality. We train both generator and discriminator using RMSprop optimizer with learning rates starting at 0.0001 at the beginning of the training process and decreasing by a factor of 0.999 after each epoch[1]. We make 8 discriminator update steps per single generator step. The batch size is 32. Model design and training is done using the TensorFlow 2.1 framework [44].

5 Results and validation

For visual evaluation of the model, in Fig. 1, we show the example pad response images from the validation dataset paired up with the ones generated with the GAN for the same track segment parameters. As can be seen from the plot, the GAN-generated images are visually similar to those obtained from the detailed simulator.

To make a more precise quantitative evaluation, we introduce a set of metrics that we profile as a function of input variables and compare between the generated and validation data. For each pad response image, we calculate the 1st order moments, i.e. the pad and time coordinates of the response distribution barycenter, the 2nd order moments, i.e. the squared widths of the response distribution and covariance between the pad and time coordinates, and the integrated amplitude. The profiles of these quantities can be seen in Fig. 3.

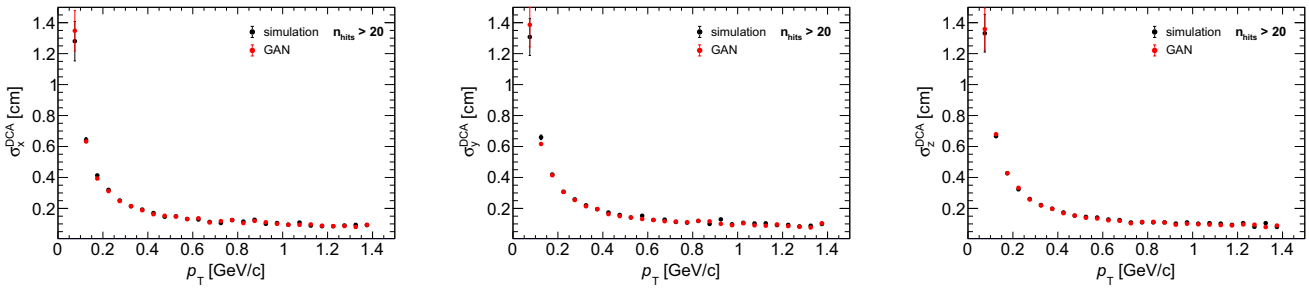
[1] By the term *epoch*, we mean a single full pass through the training dataset. When making k discriminator updates per single generator update, this implies that, per epoch, the generator is trained on $\frac{1}{k+1}$th fraction of the training data, while the discriminator – on the remaining $\frac{k}{k+1}$-th fraction.

Overall, the plots in Fig. 3 demonstrate a good agreement between the averages of the metric distributions, and also a reasonable agreement between their widths. The pad response barycenters and widths along pad row and time directions are well reproduced, which should translate to the accurate simulation of the coordinate resolution and two-track resolution of the TPC. The most notable biases may be found in the variances of the integrated amplitude distributions, as the amplitudes span several orders of magnitude and hence are particularly difficult to model precisely. Biases in the amplitude modeling can affect the dE/dx measurements, and thus the model estimates should be used with care. It should be noted that our approach is aimed to replace the expensive simulation of the electron drift and electronics response modeling, rather than track propagation. This means that Geant3 energy deposits in the detector material can be used to scale the predicted integrated amplitude and therefore account for any mismodeling of the amplitude introduced by the GAN. This is the preferred solution that is used in the tests described below.

In order to evaluate the quality of our model with reconstructed detector objects, our model is integrated into the MPD software stack. We use our model to simulate TPC pad responses for tracks from central Au+Au collisions at $\sqrt{s_{NN}} = 9$ GeV generated with UrQMD [45]. The neural network is used to simulate pad responses for the charged particles of a wide kinematic range in both long and short pads of the TPC, even though it is only trained on the responses in short pads from pions with a fixed transverse momentum. Other MPD subsystems are simulated with the detailed simulation procedures. The simulated detector responses are then fed into the standard reconstruction and track finding algorithms [46]. The comparison with the detailed simulation is then carried out for the reconstructed tracks with $|y| < 0.5$, which have at least 20 hits reconstructed in the TPC and originate from the primary pions in the event.

Figure 4a–c show the measured resolution for the distance of closest approach (DCA) to the primary vertex as a function of the transverse momentum of the particle. Figure 5 shows the momentum resolution. The figures demonstrate good agreement for the DCA resolution between our model and the detailed simulation. For momentum resolution, the agreement is good for $p > 0.9$ GeV, while at lower momentum values our model predictions result in a slightly overestimated resolution, compared to the detailed simulation.

Figure 6 shows the track reconstruction efficiency as a function of the transverse momentum and rapidity. This quantity is shown for regular and tight track selection criteria, the latter corresponding to an additional requirement on the track DCA. These plots demonstrate reasonable agreement between our model and the detailed simulation in reconstruction efficiencies.



(a) Distance of closest approach resolution along x

(b) Distance of closest approach resolution along y

(c) Distance of closest approach resolution along z

Fig. 4 Distance of closest approach resolution as a function of the transverse momentum

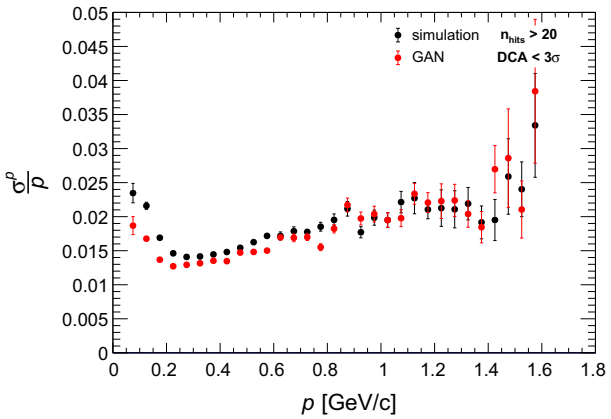


Fig. 5 Momentum resolution as a function of the full momentum

Finally, Fig. 7 shows the efficiency of matching the tracks to the signals from the Time-of-Flight (TOF) system of the MPD detector as a function of the transverse momentum and rapidity (Fig. 7a, b , respectively), and the distribution of the number of hits on track (Fig. 7c). The demonstrated agreement is excellent for the TOF matching efficiency, while the number of hits distributions are slightly inconsistent, with our model resulting in a slightly larger number of hits measured for a track. This effect is consistent with the overestimated momentum resolution and happens, as we believe, because the model is trained on the data from only the short TPC pads while utilized for the whole detector.

To demonstrate that not taking into account the difference between the long and short pads may result in the observed discrepancies, we plot distributions of deviations $\Delta x = x_{\text{reconstructed}} - x_{\text{true}}$ of the reconstructed from the true cluster coordinates for rows of short (pad row 20) and long (pad row 40) pads in Fig. 8, where x is the coordinate along the pad row direction. This should reflect the coordinate resolution of the pads. As one would expect, the GAN predictions are similar for both short and long pads, and in a reasonable agreement with the detailed simulation results for the short pads, with slight inconsistencies in the shape in the center of the peak and far in the tails. The coordinate resolution, however, is worse for the long pads, as is predicted by the wider Δx distribution from the detailed simulation, which is not captured by the GAN.

6 Discussion

As was shown in Sect. 5, the proposed model demonstrates good performance over most of the metrics considered. The barycenters and widths of the pad response distributions are well reproduced. Track reconstruction and TOF-matching efficiencies, as well as vertexing resolution, agree between the GAN and the detailed TPC simulation model predictions. We observe good agreement in a wide kinematic range, although the model is originally trained on responses from pions with a fixed transverse momentum. This can be explained by the fact that momentum and particle type mostly affect the signal amplitude rather than the shape. While the former is important for the dE/dx measurements, it is the latter that defines tracking characteristics. In fact, momentum does affect the shape of the signal through the curvature of the track, though this should be a rather little effect since the pad size is much smaller than the radius of the curvature.

Our model predictions result in an overestimated momentum resolution which goes in line with predicting more hits on track. These biases are likely to be caused by training our model on the responses from only the short pads (that have better coordinate resolution compared to the long ones) while making predictions for the whole detector. The integrated response amplitude distributions, though being reproduced well enough on average, are captured in the GAN with a slightly lower spread compared to the validation data. This may have an effect on the dE/dx measurements in TPC and would require further tuning of the GAN model, or even modifications to the architecture, e.g. a separate factorized model to only predict the integrated amplitude. Alternatively,

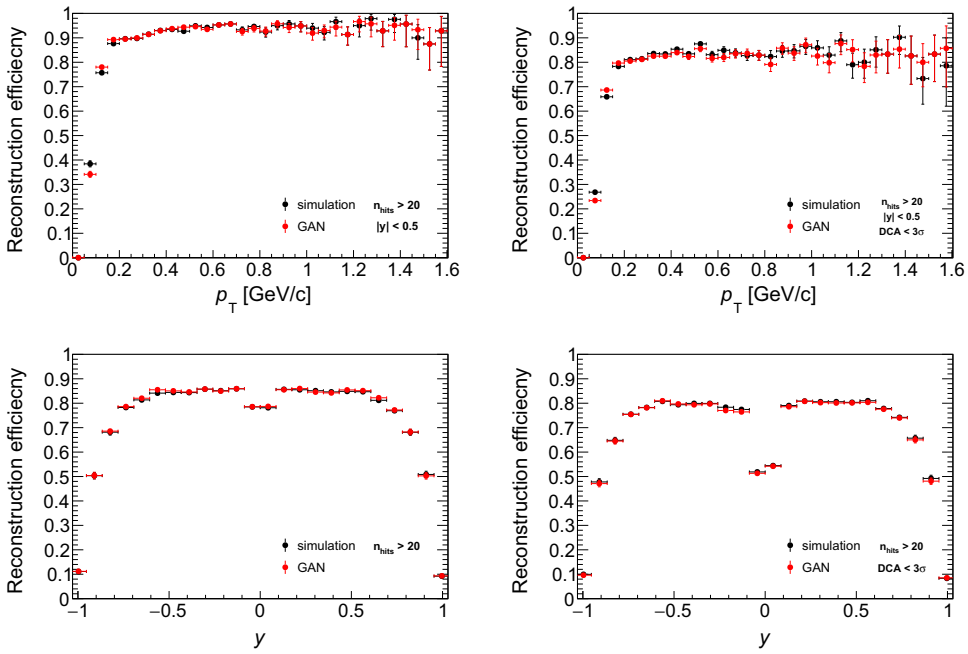
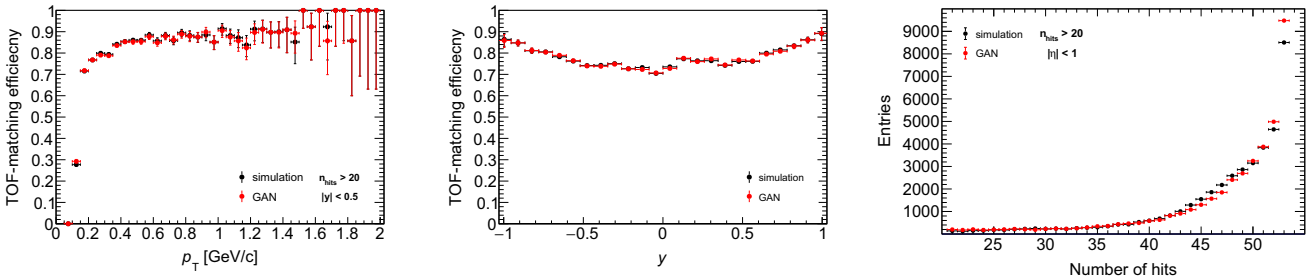


Fig. 6 Reconstruction efficiency as a function of the transverse momentum (top row) and rapidity (bottom row). Right column corresponds to the additional requirement on the reconstructed tracks quality (DCA $< 3\sigma$)



(a) TOF matching efficiency as a function of the transverse momentum

(b) TOF matching efficiency as a function of the rapidity

(c) Distribution of the number of hits per track

Fig. 7 TOF matching efficiencies over transverse momentum (**a**) and rapidity (**b**) and distribution of the number of hits per track (**c**)

one can use Geant3 energy deposits in the detector material to scale the predicted integrated amplitude and therefore account for any mismodeling of the amplitude introduced by the GAN. Investigation of the dE/dx performance, however, is beyond the scope of this work.

Along with possible enhancements in the amplitude modeling and incorporating the pad type into our model, further developments could introduce various particle types and momentum of the particle at a given track segment as additional input parameters. It is also important to evaluate the bias introduced by factorizing the responses at the adjacent pad rows.

To evaluate the performance speed-up, we run the detailed and fast TPC models on a single core of an Intel Core i7-3770K (3.50GHz) CPU, with no GPU acceleration. These tests show the GAN model integrated into the MPD software running 12 times faster compared to the detailed simulation on the central Au+Au events.

7 Conclusion

In this work, we have shown a new approach to fast simulation of TPC type detectors, based on a Generative Adversarial Network. Our model is built for the TPC detector of the MPD experiment at the NICA accelerator complex. In our approach, we split the charged particle tracks into small segments contributing to different rows of the sensitive TPC pads, and then generate the pad responses at a given row conditioned by the track segment parameters. A custom activation function is used at the generator output layer to account

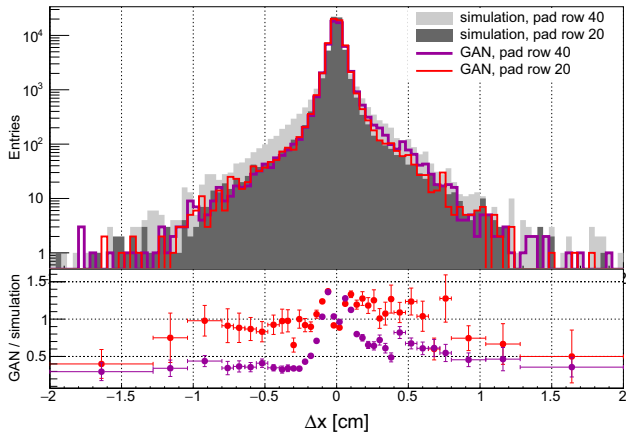


Fig. 8 Distributions of differences $\Delta x = x_{\text{reconstructed}} - x_{\text{true}}$ between the reconstructed and true cluster coordinates along the pad row direction. For the short (long) pads from the pad row 20 (40), the detailed simulation results are shown in the dark (light) gray shaded histogram, while the histogram for the GAN prediction is shown with the red (magenta) line. The ratios between the GAN and detailed simulation yields in the same pad rows are shown in the bottom part of the plot

for the discrete mode of the response distribution, caused by the noise threshold in the detailed simulation.

We have evaluated the predictions of our model by both comparing elementary response distribution characteristics, i.e. 1st and 2nd order moments and integrated amplitudes, as well as the reconstructed event properties like vertex and momentum resolution and track reconstruction efficiencies. Most of the evaluation metrics show good agreement of our model with the detailed simulation. The mismodeling of the variance of the integrated amplitude can be mitigated by scaling it with the Geant3 energy deposits in the detector material. The few inconsistencies that are seen in the reconstructed characteristics may be attributed to training the model on the responses from the short pads only while using it to predict on both short and long pads. We expect our model to be ready for use in physics tasks once trained with transverse momentum and pad type information taken into account.

Integrated into the MPD software, the proposed model runs 12 times faster on central Au+Au events, compared to the detailed simulation. This speedup translates to a factor of 2 improvement in the time spent on the overall simulation & reconstruction pipeline and therefore is sufficient at the current stage of the MPD software development.

Acknowledgements Contribution of A. Maevskiy, including development, training and evaluating the deep learning model, was supported by the Russian Science Foundation under grant agreement no. 19-71-30020. Contribution of F. Ratnikov, including the coordination of the work and the quality metric studies, was supported within the framework of the Academic Fund Program at the National Research University Higher School of Economics (HSE) in 2020–2021 (Grant no. 294715) and within the framework of the Russian Academic Excellence Project "5-100". Contribution of A. Zinchenko, including Monte Carlo simulation and performance testing of the MPD TPC, was supported by the Russian Foundation for Basic Research under Grant agreement 18-02-40060. This research was supported in part through computational resources of HPC facilities at NRU HSE.

Data Availability Statement This manuscript has associated data in a data repository. [Authors' comment: The raw pad responses the described model was trained on - https://github.com/SiLiKhon/TPC-FastSim/tree/v1.0/data/data_v4; the original PDF files of the plots used in the paper - https://github.com/SiLiKhon/TPC-FastSim/files/6778438/images.zip.]

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.
Funded by SCOAP[3].

Appendix A: Network architecture

The neural network architectures for the discriminator and generator are shown in Fig. 9.

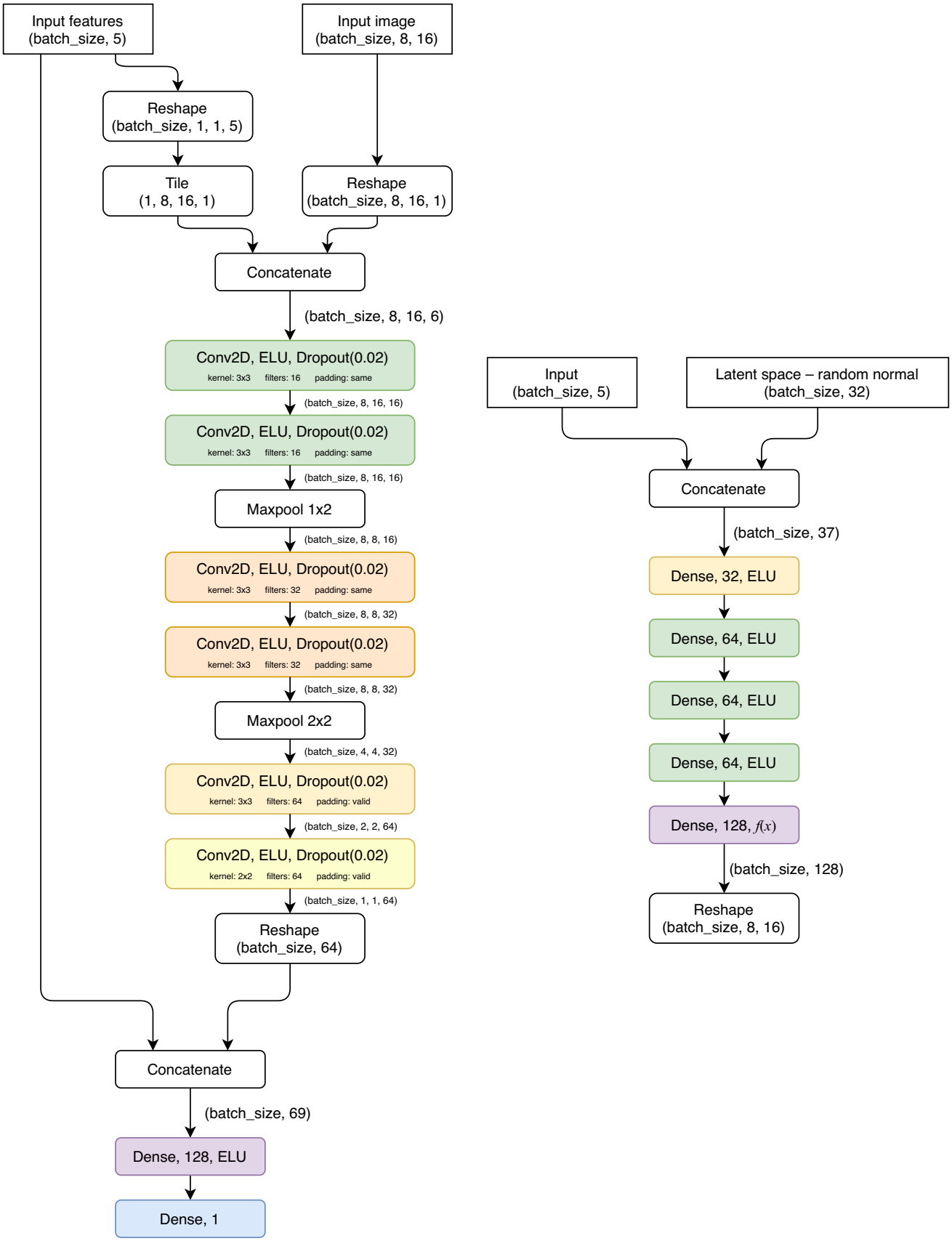


Fig. 9 Neural network architecture for the discriminator (left) and the generator (right)

References

1. A. Baranov, E. Burnaev, D. Derkach, A. Filatov, N. Klyuchnikov, O. Lantwin, F. Ratnikov, A. Ustyuzhanin, A. Zaitsev, J. Phys. Conf. Ser. **934**(1), 012050 (2017). https://doi.org/10.1088/1742-6596/934/1/012050
2. A. Boldyrev, D. Derkach, F. Ratnikov, A. Shevelev, JINST **15**(09), C09030 (2020). https://doi.org/10.1088/1748-0221/15/09/C09030. arXiv:2005.07700
3. J. Drnoyan, E. Levterova, V. Vasendina, A. Zinchenko, D. Zinchenko, Phys. Part. Nucl. Lett. **17**(1), 32 (2020). https://doi.org/10.1134/S1547477120010057
4. V. Kolesnikov, V. Kireyeu, A. Mudrokh, A. Zinchenko, V. Vasendina, Phys. Part. Nucl. Lett. **17**(3), 358 (2020). https://doi.org/10.1134/S1547477120030085
5. M. Aaboud et al., JHEP **10**, 127 (2019). https://doi.org/10.1007/JHEP10(2019)127. arXiv:1905.07163
6. G. Aad et al., Eur. Phys. J. C **80**(8), 691 (2020). https://doi.org/10.1140/epjc/s10052-020-8050-3. arXiv:1909.09226
7. J. Albrecht et al., Comput. Softw. Big Sci. **3**(1), 7 (2019). https://doi.org/10.1007/s41781-018-0018-8. arXiv:1712.06982
8. J. Chapman et al., EPJ Web Conf. **245**, 02035 (2020). https://doi.org/10.1051/epjconf/202024502035
9. G. Trubnikov, A. Kovalenko, V. Kekelidze, I. Meshkov, R. Lednicky, A. Sissakian, A. Sorin, PoS **ICHEP2010**, 523 (2010). https://doi.org/10.22323/1.120.0523
10. K. Abraamyan et al., Nucl. Instrum. Methods A **628**, 99 (2011). https://doi.org/10.1016/j.nima.2010.06.293
11. A. Averyanov et al., JINST **15**(07), C07017 (2020). https://doi.org/10.1088/1748-0221/15/07/C07017
12. D. Zinchenko, E. Nikonov, A. Zinchenko, in *Proceedings of the Conference GRID 2018* (2018), pp. 615–619. http://ceur-ws.org/Vol-2267
13. W. Lukas, J. Phys. Conf. Ser. **396**, 022031 (2012). https://doi.org/10.1088/1742-6596/396/2/022031
14. T. Yamanaka, J. Phys. Conf. Ser. **331**, 032053 (2011). https://doi.org/10.1088/1742-6596/331/3/032053
15. L. de Oliveira, M. Kagan, L. Mackey, B. Nachman, A. Schwartzman, JHEP **07**, 069 (2016). https://doi.org/10.1007/JHEP07(2016)069. arXiv:1511.05190
16. A. Aurisano, A. Radovic, D. Rocco, A. Himmel, M.D. Messier, E. Niner, G. Pawloski, F. Psihas, A. Sousa, P. Vahle, JINST **11**(09), P09001 (2016). https://doi.org/10.1088/1748-0221/11/09/P09001. arXiv:1604.01444
17. M. Andrews, M. Paulini, S. Gleyzer, B. Poczos, Comput. Softw. Big Sci. **4**(1), 6 (2020). https://doi.org/10.1007/s41781-020-00038-8. arXiv:1807.11916
18. M. Andrews, J. Alison, S. An, P. Bryant, B. Burkle, S. Gleyzer, M. Narain, M. Paulini, B. Poczos, E. Usai, Nucl. Instrum. Methods A **977**, 164304 (2020). https://doi.org/10.1016/j.nima.2020.164304. arXiv:1902.08276
19. F.A. Di Bello, S. Ganguly, E. Gross, M. Kado, M. Pitt, L. Santi, J. Shlomi, Eur. Phys. J. C **81**(2), 107 (2021). https://doi.org/10.1140/epjc/s10052-021-08897-0. arXiv:2003.08863
20. I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, (2014). arXiv:1406.2661
21. D.J. Rezende, S. Mohamed, D. Wierstra, (2014). arXiv:1401.4082
22. D.P. Kingma, M. Welling, (2014). arXiv:1312.6114
23. L. de Oliveira, M. Paganini, B. Nachman, Comput. Softw. Big Sci. **1**(1), 4 (2017). https://doi.org/10.1007/s41781-017-0004-6. arXiv:1701.05927
24. M. Paganini, L. de Oliveira, B. Nachman, Phys. Rev. Lett. **120**(4), 042003 (2018). https://doi.org/10.1103/PhysRevLett.120.042003. arXiv:1705.02355
25. M. Erdmann, L. Geiger, J. Glombitza, D. Schmidt, Comput. Softw. Big Sci. **2**(1), 4 (2018). https://doi.org/10.1007/s41781-018-0008-x. arXiv:1802.03325
26. M. Erdmann, J. Glombitza, T. Quast, Comput. Softw. Big Sci. **3**(1), 4 (2019). https://doi.org/10.1007/s41781-018-0019-7. arXiv:1807.01954
27. V. Chekalina, E. Orlova, F. Ratnikov, D. Ulyanov, A. Ustyuzhanin, E. Zakharov, EPJ Web Conf. **214**, 02034 (2019). https://doi.org/10.1051/epjconf/201921402034. arXiv:1812.01319
28. The ATLAS Collaboration, *Deep generative models for fast shower simulation in ATLAS* (2018). ATL-SOFT-PUB-2018-001, http://cds.cern.ch/record/2630433
29. P. Musella, F. Pandolfi, Comput. Softw. Big Sci. **2**(1), 8 (2018). https://doi.org/10.1007/s41781-018-0015-y. arXiv:1805.00850
30. D. Derkach, N. Kazeev, F. Ratnikov, A. Ustyuzhanin, A. Volokhova, Nucl. Instrum. Methods A **952**, 161804 (2020). https://doi.org/10.1016/j.nima.2019.01.031. arXiv:1903.11788
31. A. Maevskiy, D. Derkach, N. Kazeev, A. Ustyuzhanin, M. Artemev, L. Anderlini, J. Phys. Conf. Ser. **1525**(1), 012097 (2020). https://doi.org/10.1088/1742-6596/1525/1/012097. arXiv:1905.11825
32. K. Deja, T. Trzcinski, L. Graczykowski, EPJ Web Conf. **214**, 06003 (2019). https://doi.org/10.1051/epjconf/201921406003
33. R. Di Sipio, M. Faucci Giannelli, S. Ketabchi Haghighat, S. Palazzo, JHEP **08**, 110 (2019). https://doi.org/10.1007/JHEP08(2019)110. arXiv:1903.02433
34. S. Diefenbacher, E. Eren, G. Kasieczka, A. Korol, B. Nachman, D. Shih, JINST **15**(11), P11004 (2020). https://doi.org/10.1088/1748-0221/15/11/P11004. arXiv:2009.03796
35. A. Hariri, D. Dyachkova, S. Gleyzer, (2021). arXiv:2104.01725
36. J. Gui, Z. Sun, Y. Wen, D. Tao, J. Ye, (2020). arXiv:2001.06937
37. J. Arjona Martínez, T.Q. Nguyen, M. Pierini, M. Spiropulu, J.R. Vlimant, J. Phys. Conf. Ser. **1525**(1), 012081 (2020). https://doi.org/10.1088/1742-6596/1525/1/012081. arXiv:1912.02748
38. K.T. Matchev, P. Shyamsundar, (2020). arXiv:2002.06307
39. A. Butter, S. Diefenbacher, G. Kasieczka, B. Nachman, T. Plehn, Sci. Post Phys. **10**, 139 (2021). https://doi.org/10.21468/SciPostPhys.10.6.139. arXiv:2008.06545
40. A. Averyanov, et al., *Time Projection Chamber for Multi-Purpose Detector at NICA* (2019). Technical Design Report (rev.07). http://mpd.jinr.ru/wp-content/uploads/2019/01/TpcTdr-v07.pdf
41. V. Kolesnikov, A. Mudrokh, V. Vasendina, A. Zinchenko, Phys. Part. Nucl. Lett. **16**(1), 6 (2019). https://doi.org/10.1134/S1547477119010084
42. M. Arjovsky, S. Chintala, L. Bottou, (2017). arXiv:1701.07875
43. I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. Courville, (2017). arXiv:1704.00028
44. M. Abadi, et al., (2016). arXiv:1603.04467
45. M. Bleicher et al., J. Phys. G **25**, 1859 (1999). https://doi.org/10.1088/0954-3899/25/9/308. arXiv:hep-ph/9909407
46. K. Gertsenberger, S. Merts, O. Rogachevsky, A. Zinchenko, Eur. Phys. J. A **52**(8), 214 (2016). https://doi.org/10.1140/epja/i2016-16214-y